AR-010-309

DTIC

Computer Support for Schema Integration

C. A. Ewald

DSTO-TN-0072

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Computer Support for Schema Integration

C.A. Ewald

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TN-0072

ABSTRACT

Currently, as part of an organisation-wide move for greater interoperability and reduction of duplication, Defence is examining a number of existing information systems in order to make them work more closely together. Schema integration is the merging of different database design specifications which have commonality. In this report, we examine support for schema integration provided by commercial off the shelf (COTS) software products,in particular computer aided software engineering (CASE) tools, and suggest desirable features which current products do not support. We examine which of the two products tested provides cost-effective support for schema evolution. The two products chosen for examination are InfoModeler version 2 (chosen for its support of the rich object-role modelling methodology, and natural language interface) and ERwin Version 2.6.1 (the market leader both in Australia and the U.S.A.).
It appears that no existing product provides a full schema merge capability, and this would not be expected, as schema integration needs to involve human creativity. However, some tasks within the integration process lend themselves to automation, and it is worthwhile examining tools to perform these tasks.

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-010-309
September 1997

APPROVED FOR PUBLIC RELEASE

Computer Support for Schema Integration

Executive Summary

Currently, as part of an organisation-wide move for greater interoperability and reduction of duplication, Defence is examining a number of existing information systems in order to make them work more closely together. Schema integration is the merging of different database design specifications which have commonality. Commercial off-the-shelf (COTS) tools for schema integration are often expensive, as are custom-developed solutions. Like database design, schema integration is a process which requires creative human input, so full automation of the process is not expected. While no current COTS software products provide all the features considered desirable in a product to support the integration process, a lot can be done with state-of-the-art tools. In this report, we examine two such tools, showing that at least one has many features which provide significant benefit to the integrator. The tools chosen for this evaluation were InfoModeler Version2, and ERwin Version2.6.1. InfoModeler was selected for its support of the very expressive object role modelling language FORML, and for its natural language interface. ERwin is the market leader, both in Australia and the USA .
We considered features helpful in schema integration, and used in a different fashion from that expected in traditional schema design. These were schema comparison, reverse engineering, and integrity checking. Overall, InfoModeler was found to be more suitable for schema integration.

A number of features useful to integration were missing from both tools. In particular, relational integrity checking was not performed. We suggest that the integrated schema be checked for cycles and redundant functional dependencies. Simple and fast algorithms are available for these processes. A method of generating sample populations which illustrate a given set of constraints, and can be used to highlight a normally "hard to find" class of conflicts is also suggested.

Authors

Cathy Ewald
Information Technology Division

Cathy Ewald was awarded a PhD in Computer Science from The University of Queensland, Brisbane in 1996. She has conducted research into object-relational and nested relational databases, and into schema integration at the University of South Australia and the University of Queensland. She joined DSTO in January 1997, and is continuing to address research issues related to making distinct databases interoperate.

Contents

1. Introduction

Currently, as part of an organisation-wide move for greater interoperability and reduction of duplication, Defence is examining a number of existing information systems in order to make them work more closely together. Schema integration is the merging of different database design specifications which have commonality. A computer based tool for schema integration should not interfere with the human creativity essential to such work, but should provide as much support as possible for the humans involved in the process. In this report, we examine two COTS database design tools, and compare their strengths and weaknesses for the schema evolution problem. We also describe additional features which neither tool supports, and which we suggest would be beneficial. The tools chosen for this experiment were ERwin Version2.6.1 and InfoModeler 2. The tools were tested by altering sample schemata which came with the tool, and integrating the altered and new versions. Thus, the schemata were structurally very similar, with constraint and relationship differences. The tools were compared with respect to features used in integration. The main features considered were schema comparison(that is the identification of similarity or difference), reverse engineering (extraction of a schema from a working database or from data definitions in SQL or similar), and integrity checking (the detection of constraint conflicts and other problems introduced by integration). For each of these features, a description of the performance of each product is given, then the support of the products for the feature is rated on a scale of one to ten. Both technical performance and suitability for the Defence environment are taken into account. For example, InfoModeler performs re-engineering well, but receives a lower score than ERwin for that task because it requires a connection to the on-line database. This raises issues of security and access rights in the Defence environment. Some other factors are also briefly examined. We then suggest some integrity checking algorithms based on relational database theory which would also be beneficial, though they are not included in either tool.

2. Schema Comparison

By schema comparison, we mean all those features which aid the user in finding similarities and differences between entities, relationships and constraints on the schema.

We review the way in which each product supports this operation.

The standard edition of ERwin, on which our testing was carried out, has no custom facilities for comparisons of schemata. However, the add-on product Model Mart provides a "Complete Compare" function which detects and reports on differences in entity and relationship definition, and specification of a limited variety of constraints (participation, and keys). Complete Compare is primarily designed to reconcile different versions or views of the same schema, but would be helpful in identification of constraint conflict, and overlap between schemata, provided that they were substantially similar. It assumes that all naming conflicts have been resolved. That is , items with different names are taken to be different items. Model Mart has a merge feature which actually combines schemata, but once again these need to be substantially similar. In addition, Model Mart provides a sophisticated tool for configuration management. This includes good support for version management and tracking , but these facilities would have limited application to the schema integration problem. It should also be noted that Model Mart is an optional add-on, which must be purchased at extra cost.

InfoModeler does not require add-on software to support integration, with the configuration management and comparison features being a part of the standard product. Schemata to be integrated or related in some way are explicitly gathered by the user into a project. The tool attempts to reconcile schemata within a project, highlighting cases where any item is multiply defined. One can then mark some version(s) of the items in question as "external" , meaning that they are allowed variations from the official definition, or rename them using a sophisticated name space maintenance feature. Once again, this is not an integration or merging tool, but a function similar to Complete Compare. As with Complete Compare, the tool cannot cope automatically with naming conflicts. Objects which have the same name are highlighted by the tool. However, similar objects with different names are ignored. Models within a project are still regarded as independent models. However, creation of a "dictionary" merges them into a single new schema. The old schemata are left unchanged, though linked with the dictionary so that changes in a source schema can be propagated to the dictionary and vice versa.

While is this not a "virtual table", and does not create the resultant technical problems, it is a similar concept when seen from a conceptual viewpoint. Human intervention is required in this process.

Comparison Rating for ERwin : 4/10
Comparison Rating for InfoModeler 8/10

3. Reverse Engineering

Reverse engineering is the process of defining a schema from a database implementation.

ERwin was found to give satisfactory performance in reverse engineering, from either a data definition language script or an active database. Non-relational legacy systems such as A/REV and COBOL cannot be re-engineered using ERwin, nor can object-based or object relational systems.

In InfoModeler , a reverse -engineering facility exists, but requires an active database connection. This is a disadvantage for some defence work in which users are prepared to give access to metadata, but not grant connection rights to databases. "Dummy" databases can be constructed from metadata, but when operating in a heterogeneous environment, this requires access to a large number of different database management systems. The issue in our environment is that users of working, on- line systems are understandably reluctant to allow meta-data managers to access, and run tools on, their working system. It is regarded as much safer to simply allow access to an SQL "build file" which describes the structure of the database. This avoids the need for these people to have power over working systems, and also reduces the expense in purchases of a number of different database management programs. Once again, only relational and common desktop databases such as MS Access systems can be re-engineered. However, a new release of InfoModeler is now available which supports the full ranges of SQL 3 constructs, thus allowing most extended data types to be understood by the tool. Users even have the facility to define their own types.

Reverse engineering Rating for ERwin : 8/10
Reverse engineering Rating for InfoModeler 6/10

4. Integrity Checking

Integrity checking is the process of detecting semantic conflicts due to integration.

Neither tool provides a great degree of support for this process. The main features for constraint conflict detection have been discussed in the previous section. Recall that Model Mart detects some conflicts in its Complete Compare, and InfoModeler when a schema is added to a project or explictly checked by the user. Model Mart was not tested due to cost considerations.

While the tools do not have specifically tailored features for this task, InfoModeler was found to have generic features which could be adapted to support this process. Firstly, in addition to the entity-relationship (ER) modelling language supported by ERwin and other tools, Info Modeler gives access to a more expressive Object-Role Modelling (ORM) language, which allows a wider range of constraints to be expressed in the model. ER is the most widely used data modelling language in Defence, as well as in Australian and American industry. Constraints in ORM which are not supported by ER include, reflexive, transitive and symmetric constraints on circular relationships , domain range constraints, and subset equality and exclusion constraints between sets of entities playing roles related in some sense. Thus, by definition the constraint report of InfoModeler is potentially more powerful. InfoModeler can additionally identify some cycles, a large number of mistakes or redundancies related to subtyping, many common data modelling mistakes, lack of a unique identifier for an entity and some types of constraint conflict. Relationship errors, such as a relationship accidentally reversed are detected when checking is done, and the tool does not allow relationships to be included "in the wrong direction". A subtype that is not defined in terms of its supertype will be reported. It also detects cases where any data item has more than one definition or more than one different set of constraints.

None of the checks performed by either of these tools require access to populations (instances). However, InfoModeler allows the user to input a sample population for every "fact". Certain constraints, namely those related to keys (unique identifiers) can also be inferred from these user supplied information examples. Thus, if the constraint is not consistent with the examples, the tool detects this. The tool has the capability to suggest constraints of this type based on a set of examples. This facility is restricted to uniqueness (key) constraints, which can be inferred from data using algorithmic techniques. This facility is not intended to check the quality of data, rather as a "reality check" on the design. That is , it helps users to decide if the constraints they have specified exclude data which they might wish to include or fail to exclude situations forbidden in the real world. It is equally applicable to the design of a new schema as to integration. The user is offered the change to alter uniqueness constraints which are suggested by the system.

ERwin performs some checking related to design errors, for example identification of some cycles on the schema. Only structural checks such as a check for cycles involving entities and subtypes are performed.

Checking Rating for ERwin 4/10

Checking Rating for InfoModeler 8/10

5. Other Factors

In this section, the general features of the tools are examined. Clearly, users want a tool that they can use easily, in addition to having features for schema integration. The general features of the packages when used for ER modelling are comparable , to the extent that the choice of product for conventional ER modelling is largely a matter of taste. Both have good user interfaces, the ability to "browse" a schema for a particular construct, searching tools, the ability to define complex colour schemes and some checking facilities. InfoModeler , as mentioned earlier , has more ability to analyse and check a schema and comes with advanced reporting capabilities. It also provides a user interface which encourages the users to properly document the model. When using FORML, the user can perform modelling almost entirely in natural language. ERwin runs on a wide variety of hardware platforms and operating systems, while InfoModeler only runs on Windows platforms. Data can be imported into InfoModeler from ERwin, when a .dll file to perform data conversion is installed. This file is supplied with the current release, and is freely available. ERwin cannot directly read InfoModeler files. Reports and schemata from InfoModeler can be exported as RTF files, for inclusion in written reports. InfoModeler can generate, but not read, a wide range of SQL and PC based database script files. As mentioned earlier, reverse engineering can be performed if connected to the database. ERwin can generate and read most SQL and PC database script files, providing a way of interchanging data. ERwin also provides a wide range of report formats, as does InfoModeler. ERwin reports may be exported as text files delimited by commas or tabs, to allow input into other programs such as word processors. RTF format is not provided.

Both stand alone products cost less than $6000, with InfoModeler's estimated recommended retail price per seat being $4600, and $5499 for ERwin. A cheaper desktop version of InfoModeler ($800) is also available. This version has the full range of modelling features, but can only reverse-engineer from, and generate code for, desktop PC databases. Five copies of ERwin, with Universal Directory and Model Mart cost around $40 000.

6. Features not Included in Either Tool

As has been mentioned earlier, neither tool provides a full integration function. As mentioned in the previous section, current COTS products provide only limited support for integrity checking in the context of schema integration. We suggest that some algorithmic checking procedures based on relational database theory be implemented. Since schema diagrams identify keys, functional dependency (FD) constraints can be extracted from any schema and mapped to a graphical representation. In conceptual data models, most FDs are key constraints, that is based on the unique identifier of an entity. Functional dependency graphs are defined and explained in [Yang86]. In [Ewald96] and associated papers (E093,E094,E095) simple graph search and comparison operations are then used to detect conflict, redundancy and inconsistency on evolving schemata. The classical synthesis algorithm [Maier 83] can also be applied to remove redundancy or to produce a well-designed relational database from a conceptual specification.

Cycles on a conceptual schema may result from addition of a redundant dependency and from conflict between existing and new dependencies. A redundant dependency is one which can be derived from the others on the schema using a set of formal axioms. For this reason, it is often beneficial to have a human examine all cycles on a schema to ensure

that none result from such schema problems. A "loop" on the conceptual schema may result in the same information being stored within a local schema in multiple ways. Addition of new information to a schema may also create such a loop. Conflicting dependencies may be added during integration, resulting in a graph which contains two representations of the dependency relationship between a given pair of attribute sets.

All these situations can be detected by performing simple graphical tests on the FD graphs created from the schemata under consideration. Comparison of current and previous graphs is used to detect conflict, while depth first search identifies cycles. These quick screening tests report any potential problems to a human designer, who can then use the synthesis algorithm or conceptual re-design techniques to eliminate problems.

The above techniques still detect only a limited set of constraint conflicts, namely those involving functional dependency type constraints. These are the constraints which can be checked without reference to sample schema populations. A further level of checking which considers a larger range of constraints, and examines the potentially complex interactions between constraints is recommended. This is not done by either of the tools examined above. From a functional dependency graph, we can generate a list of the dependencies which are known not to hold. These are known as potential violations, and a formal theory is presented in [Ewald96]. Based on these, algorithms have been developed to detect constraint conflict. This is done by constructing constraint patterns, using an algorithm which halts if a pattern cannot be found. These algorithms also generate small, meaningful sets of information examples which illustrate the features of the set of constraints under consideration. These are particularly useful with the ORM modelling techniques, which encourage interaction with users by means of information examples. The InfoModeler tool allows the designer to enter examples of data elements at design time, and can infer some constraints based on such examples. Conflict between functional, set inclusion, set exclusion and mandatory constraints may be detected by these algorithms. Real information examples from a repository or a relational database may be retrieved to provide output in a form with which the users of the system are familiar.

7. Conclusions

Current COTS products provide useful support for schema integration. In this report, the support of ERwin 2.6.1 and InfoModeler 2 for the basic activities of the integrator is assessed.

The table below summarises the results of the investigations, considering a version of ERwin without Model Mart extensions. Note that Model Mart supports relevant features, but at a cost. Graphical user interface, and support for ER modelling are similar in the two products. InfoModeler's additional ORM features complement the ER modelling methodology, and provide a more detailed view of the data model. It is possible to completely convert from ER to ORM, however a recommended approach is to use both methods, presenting the ER view as an abstraction when a fully detailed view is not needed. This also allows communication with domain experts in natural language, and with experienced modellers using whichever graphical notation they prefer. The InfoModeler product supports conversion of models between ER and ORM forms and detailed integrity checking before conversion.

	Comparison	Reverse Engineering	Integrity Checking	Graphical FD checking	Generation of Examples
ERwin	4	8	4	None	None
InfoModeler	8	6	8	None	None

As can be seen, existing software can help with schema integration, but neither product examined could be considered to be a full schema integration tool. Some easy-to-implement features which would be beneficial in such a tool are suggested. In particular, checking for cyclic key constraints, removal of redundancy, and generation of suitable information examples are beneficial and computationally cost-effective. The examples generated are artificial, but an alternative is to use algorithms to select a small amount of relevant data from a larger repository of examples (either real data or examples given by users). That is, the algorithms as originally developed construct symbol "constraint patterns" using semantically meaningless symbols. However, these are then replaced by matching data from a repository, which is more meaningful, especially to domain experts from areas other than information technology.

8. Acknowledgment

The author would like to thank the DSTO reviewer, Mr Conn Copas, for his helpful comments on a draft version of this report.

9. References

1. [Ewald 96] Catherine A. Ewald. *Foundations of Conceptual Schema Evolution* PhD Thesis, Department of Computer Science, The University of Queensland, Feb. 1996.
2. [E093] C.A. Ewald and M.E. Orlowska . A Procedural Approach to Schema Evolution. *Proceedings of the 5th International Conference on Advanced Information Systems Engineering* pages 22-28, Springer Verlag (LNCS), June 1993
3. [E094] C.A. Ewald and M.E. Orlowska An efficient graphical algorithm for incremental conceptual schema evolution *Proceedings of the First International Conference on Object Role Modelling, 1994.*
4. [EO95] C.A. Ewald and M.E. Orlowska Meaningful Significant Information Examples for Schema Design and Evolution Technical Report 340, Department of Computer Science, The University of Queensland, 1995.
5. [Maier 83] Maier, D. *Theory of Relational Databases* Computer Science Press, 1983.
6. [Yang 86] Yang,CC. *Relational Databases* Prentice Hall, Englewood Cliffs, NJ, 1986

Computer Support for Schema Integration

C.A. Ewald

(DSTO-TN-0072)

DISTRIBUTION LIST

AUSTRALIA

DEFENCE ORGANISATION

Task sponsor:

Director General C3I Development	1

S&T Program

Chief Defence Scientist	)	
FAS Science Policy	)	1 shared copy
AS Science Corporate Management	)	
Director General Science Policy Development		1
Counsellor, Defence Science, London		Doc Control Sheet
Counsellor, Defence Science, Washington		Doc Control Sheet
Scientific Adviser to MRDC Thailand		Doc Control Sheet
Director General Scientific Advisers and Trials	)	(1 shared copy)
Scientific Adviser - Policy and Command	)	
Navy Scientific Adviser		1 copy of Doc Control Sheet and 1 distribution list
Scientific Adviser - Army		Doc Control Sheet and 1 distribution list
Air Force Scientific Adviser		1
Director Trials		1

Aeronautical & Maritime Research Laboratory

Director	1

Electronics and Surveillance Research Laboratory

Director	1
Chief Information Technology Division	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Computing Systems	1
Research Leader Command, Control and Communications	1
Executive Officer, Information Technology Division	Doc Control Sheet
Head, Information Architectures Group	1
Head, Information Warfare Studies Group	Doc Control Sheet
Head, Software Systems Engineering Group	Doc Control Sheet

Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Advanced Computer Capabilities Group	Doc Control Sheet
Head, Computer Systems Architecture Group	Doc Control Sheet
Head, Systems Simulation and Assessment Group	Doc Control Sheet
Head, Intelligence Systems Group	Doc Control Sheet
Head, CCIS Interoperbility Lab	Doc Control Sheet
Head Command Support Systems Group	1
Head, C3I Operational Analysis Group	Doc Control Sheet
Head Information Management and Fusion Group	1
Head, Human Systems Integration Group	1
Task Manager John Mansfield	1
C. Ewald	1
Publications and Publicity Officer, ITD	1
DSTO Library and Archives	
Library Fishermens Bend	1
Library Maribyrnong	1
Library Salisbury	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
Capability Development Division	
Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Intelligence Program	
Defence Intelligence Organisation	1
Corporate Information Program	
Director General Information Policy and Plans	1
Director Metadata Management	3
Corporate Support Program (libraries)	
OIC TRS Defence Regional Library, Canberra	1
Officer in Charge, Document Exchange Centre (DEC),	1
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1
Universities and Colleges	
Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Senior Librarian, Hargrave Library, Monash University	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations

INSPEC: Acquisitions Section Institution of Electrical Engineers	1
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts	1
Documents Librarian, The Center for Research Libraries, US	1

Information Exchange Agreement Partners

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES 10

Total number of copies: 64

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Computer Support for Schema Integration

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

C.A. Ewald

5. CORPORATE AUTHOR

Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury SA 5108 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TN-0072	AR-010-309	Technical Note	September 1997

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
N9505/13/123	ADF 96/182	DGC3ID	10	6

13. DOWNGRADING/DELIMITING INSTRUCTIONS

-

14. RELEASE AUTHORITY

Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Database Integration; Database design; interoperability

19. ABSTRACT

Currently, as part of an organisation-wide move for greater interoperability and reduction of duplication, Defence is examining a number of existing information systems in order to make them work more closely together. Schema integration is the merging of different database design specifications which have commonality. In this report, we examine support for schema integration provided by commercial off the shelf (COTS) software products,in particular computer aided software engineering (CASE) tools, and suggest desirable features which current products do not support. We examine which of the two products tested provides cost-effective support for schema evolution. The two products chosen for examination are InfoModeler version 2 (chosen for its support of the rich object-role modelling methodology, and natural language interface) and ERwin Version 2.6.1 (the market leader both in Australia and the U.S.A.).

It appears that no existing product provides a full schema merge capability, and this would not be expected, as schema integration needs to involve human creativity. However, some tasks within the integration process lend themselves to automation, and it is worthwhile examining tools to perform these tasks.